
METACONFORMER
MODULAR MIDI PROCESSOR
USER
MANUAL
USB
STATE/DATA
CFG
RUN
RST
METACONFORMER
MODULAR MIDI PROCESSOR

# CONTENT

# GENERAL OVERVIEW

Metaconformer is a modular MIDI processor designed to expand the capabilities of musical instruments and to simplify the interaction between a musician and their MIDI instruments. It offers new ways of playing music through innovative and groundbreaking MIDI message routing capabilities between multiple instruments.

The MIDI protocol is an efficient engineering solution used reliably for decades by musicians. However, in many situations it does not take into account the musical context, which is essential for the artist. Metaconformer changes this. Located between the MIDI controller and your instruments, it coordinates human ideas with the language of hardware.

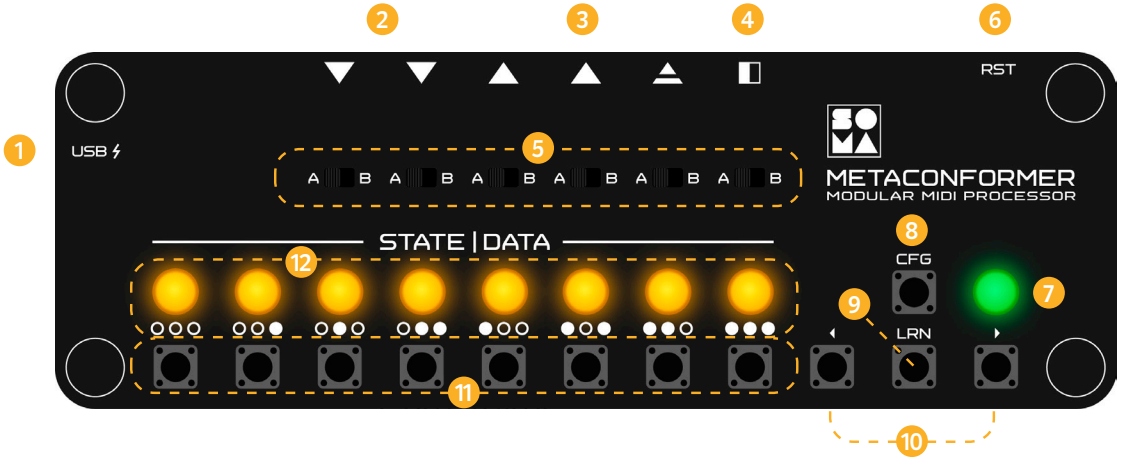
Metaconformer's capabilities provide new ways of playing instruments, by combining MIDI instruments and exploring their interaction. You will be able to play multiple instruments at the same time as if they were part of one larger instrument. Metaconformer also opens up the possibility of sound control through pressing the keys on your MIDI keyboard, instead of having to reach for the knobs on your gear. This will allow you to discover and unlock new facets and possibilities of your favorite instruments!

The idea behind the Metaconformer is to give musicians a new way of looking at their instruments and what else they can do with a little creativity in MIDI.

SOMA is closely related to the idea of organismic synthesizers. Each of our instruments aims to give the musician the experience of interacting with a system in which several independent modules can influence each other and create more than the sum of its parts.

Metaconformer is no exception. It breaks down at least two barriers that have stood in the way of all MIDI musicians for decades. It establishes a dialogue between different instruments, breaking the barrier of parallel universes of MIDI channels. And Metaconformer also allows you to enter the values of synthesis parameters not by turning knobs on your synths, but by pressing keys on your MIDI keyboard, breaking the barrier of the one-dimensional universe of notes and creating a multi-dimensional universe of parametric chords.

## HARDWARE




1 **USB power connector.** Through the same connector, the firmware is updated. Read more in the Firmware Update section.

2 **Two MIDI inputs**, messages from which are combined into a single stream.

3 **Three MIDI outputs.**

4 **Hybrid output**, that works in two modes: MIDI output or sync signal output with a controlled divider and polarity. Read more in the Hybrid output section.

5 **The mini-TRS MIDI connectors** are equipped with a type A and B switch to make the Metaconformer compatible with any instrument. It is recommended to connect only stereo mini-jack TRS to MIDI cables.

This switch can act as an hardware on/off switch for each MIDI connector.

6 **RST button** to reboot the device.

7 **The green LED** indicates which menu or mode the user is in. Solid light indicates the Learn menu. No green light means Play mode. Blinking light means Global Configuration Menu.

8 **The CFG button** is responsible for switching to the Metaconformer Global Configuration menu. In this menu, you can adjust the hybrid output, MIDI processor input channel, and other mode-specific settings. Read more in the section Global Configuration Menu.

9 **The LRN button** is responsible for switching to the Learn menu of the selected MIDI processor cell. See the tutorials for operating modes for details.

10 **The ◂ and ▸ button** are responsible for the cyclic movement of the Metaconformer cursor. The cursor is responsible for selecting one of the eight MIDI processor cells and for selecting a parameter in the Global Configuration Menu and the Learn menu.

11 **Cell buttons** are responsible for temporarily muting processor cells.

12 **The STATE/DATA LEDS** function as an 8-bit display that provides information about the state of the device in each operating mode.

## FUNCTIONAL DESCRIPTION

Metaconformer can be used as a MIDI splitter. Messages from each MIDI input are collected separately, and then combined into a single stream.

To avoid various unexpected situations and conflicts, it is recommended not to send messages from the same MIDI channel to both MIDI inputs of the Metaconformer.

*After the messages are assembled, they are divided into two groups:*
1. Messages from the input channel of the MIDI processor
2. Messages from other channels.

Messages from the second group are immediately sent to all MIDI outputs of the Metaconformer. In Play mode, the green LED flashes as notes pass through Metaconformer. One blink for noteOn, one blink for noteOff.

Messages from the first group are sent to the **MIDI Processor**, where, after being processed, they are also sent to the MIDI output. See the Combiner Tutorial, Splitter Tutorial, Translator Tutorial for details on how the MIDI processor works in the different modes.

## MIDI PROCESSOR.
## BRIEF DESCRIPTION OF OPERATING MODES

This processor consists of eight identical cells, the functionality of which depends on the operating mode of Metaconformer.

Metaconformer can operate in one of three processing modes:

The **Combiner** does the same as any polyphonic synthesizer when it chooses which voice will play the note coming from the MIDI input channel. But only independent instruments act as voices.

This mode allows you to combine several monophonic synthesizers or samplers into one instrument. The order in which the notes are activated determines the final timbre of the sound. The process is similar to playing an entire orchestra because each voice has its own unique dynamics and can be activated independently of the others.

Voices can be muted to hear how the chord will sound, made up of different combinations of timbres.

Also, by moving the cursor of the active cell, you choose from which voice the Metaconformer starts looking for a free cell. More details in Combiner Tutorial.

**Splitter** allows you to play multiple instruments on the same MIDI channel by splitting the

keyboard into zones or ranges. This is useful in situations where the instrument only spans a limited range of notes, such as samplers or drum machines.

This makes it possible to place up to eight instruments on a single MIDI keyboard, based on their role in the mix, so you can conveniently play them all at the same time and hear the interaction of timbres without having to switch keyboard channels or use multiple keyboards.

The ranges may overlap.

Each range has 16 different transposition options. Including the ability to invert notes and collapse the entire range into one of the extreme notes of the range.

Any range can be temporarily muted.

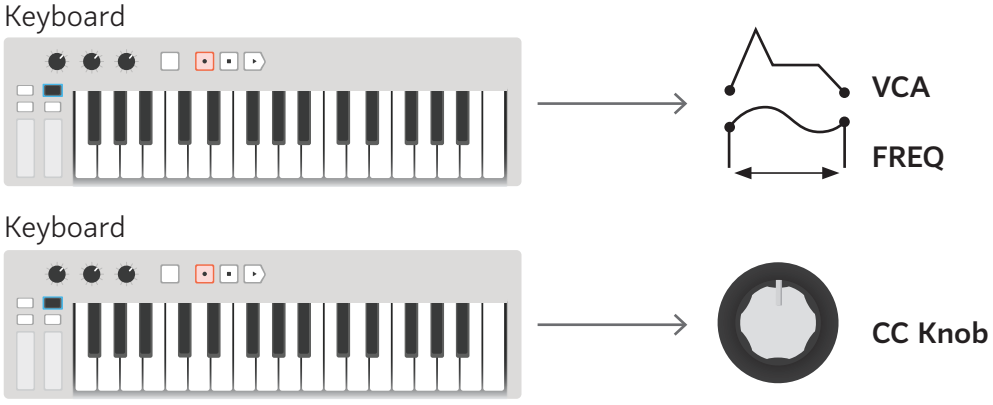
See the Splitter Tutorial for details.

**Translator** uses the keys on a MIDI keyboard to affect sound parameters in your synths's sound engine. Instead of turning knobs, play your keyboard to change the sound.

Each Metaconformer cell in this mode translates MIDI notes from the input MIDI channel into CC messages that control parameters of the connected synth.

To use this mode effectively, you must have access to two MIDI channels at the same time. The first channel is responsible for selecting the note and activating the VCA envelope of your synth, otherwise you won't hear anything. The second channel is responsible for selecting the value of the synth parameter through the number of MIDI note presses.

Of course, you can use two keyboards, but it's easier to use a second Metaconformer in




Splitter or Combiner mode for this. In this case, Metaconformer can provide you with as many as eight keyboards with simultaneous access.

Each Translator cell can be temporarily muted by pressing its cell button. In this case, the notes from its input channel will go to the output without changes.

See Translator Tutorial for details.

Each mode of operation (including those that will be released with future firmwares) has a separate tutorial that describes in detail the setup process, usage ideas and all the features of this mode. Additionally, visit our Instagram @somasynths and Youtube to see more examples of the use of Metaconformer, as well as the rest of our instruments.

## CONNECTION

**1.** Connect a MIDI controller to one of Metaconformer's MIDI inputs.
**2.** Connect at least one synthesizer to the MIDI output of the Metaconformer
**3.** Connect USB-mini cable to power on.
**4.** Metaconformer turns on and enters the waiting state. Now you need to select one of the operating modes.
**5.** Rotate any CC parameter on your MIDI keyboard or controller, and the Metaconformer cursor will start to follow the rotation.
**Splitter** — 
**Combiner** — 
**Translator** — 
**6.** After the cursor stops, the Metaconformer will start in the selected mode. The previously saved settings for this mode will be loaded.


If you just press CFG instead of rotating CC in step 5, the Metaconformer will start in the mode it was in before the restart.

## FIRMWARE UPDATE

Metaconformer is an open system. Users can create their own firmware with any functionality and upload it to the device. The process of updating the firmware should be done with care and users should not upload unverified firmware into its memory.

*Follow these steps to update the firmware:*
**1.** Connect Metaconformer to the computer via USB.
**2.** While the CFG button is held, press the RST button. Metaconformer will start in flash drive mode and will be detected as such on your computer, and all its LEDs will turn on.
**3.** Download the firmware from the SOMA website (https://somasynths.com/metaconformer/firmware) and copy it to Metaconformer.
**4.** Rename the file to be called blink.bin
**5.** Restart Metaconformer by pressing RST. It will start already flashed.
**6.** You can now disconnect Metaconformer from your computer.

## USER INTERFACE

Metaconformer is made to be not only useful in practice, but also easy to use. Instead of text and numbers on a matrix display, it communicates with the user through the sound of your instruments and a minimum of information on the 8-bit STATE/DATA indicator.

When playing musical instruments in different situations, this orientation method is much more convenient, because it uses not only the visual channel, but also the auditory channel, and also consumes less mental resources on the part of the user. Anyone who has ever worked with MIDI will appreciate this.

Metaconformer knows what is between the MIDI controller and the musical instrument, so it actively uses it and turns it into an advantage.

Setting up cells is fast, simple and intuitive. Metaconformer takes all information from messages sent from your MIDI controller. It's like a dialogue.

To set up a voice in Combiner mode, you need to press just one note while in the Learn menu. Metaconformer will remember the MIDI channel of that note and set up the cell.

To set the range in Splitter mode, you need to press two notes while in the Learn menu. Metaconformer will remember their MIDI channel, and the values of these notes will indicate the boundaries of the range.

To set up a cell in Translator mode, while in the Learn menu, turn the MIDI controller knob corresponding to the desired parameter, then press a note on the MIDI channel that will be translated. Metaconformer will remember the CC number of the parameter and its MIDI channel, as well as the MIDI channel from which the notes should be translated.

Details about the operation of all modes of Metaconformer are found in the respective tutorials for these modes.

You don't need a computer to set up Metaconformer, just a MIDI controller is enough.

Instead of storing a lot of static patches, which can easily get confusing, Metaconformer offers a flexible system that you set up once with your instruments, and then only need to adjust slightly according to the situation and your ideas.

Metaconformer is like a Swiss Army knife. He does not know how to do everything, but he is always at hand. And what he can do, he does excellently.

## 8-BIT STATE/DATA DISPLAY

The simple 8-bit display in the spirit of ancient computers from the 1960s is capable of displaying binary coded values, or cell states in on / off format. Such a display minimally distracts from playing your instruments, while being able to convey all the necessary information.

In Play mode, the display shows the state of cells as active / inactive, the position of the cursor and active cells. In global configuration and learn mode, it shows binary coded parameter values.

For details about displaying information from the Learn menu and Global Configuration, see the Tutorials of the respective operating modes and the Feedback and General Function Tutorial.

Metaconformer is designed so that you don't need to interact with binary coded numbers at all if you prefer, as all feedback is available in the form of sounds from the connected instruments. However, Metaconformer still outputs a minimum of information in binary form. You can get familiar with binary numbers in the Extras section to get the most out of Metaconformer feedback.

## CONCLUSION

The name is derived from the words meta — between, and conform. A device that is placed between the musician and the instrument.

A device that understands the desires of the musician and is able to translate these desires into the language of hardware.

A device that will help you access new possibilities that were previously hard to reach or hidden.

A device that brings your instruments closer.

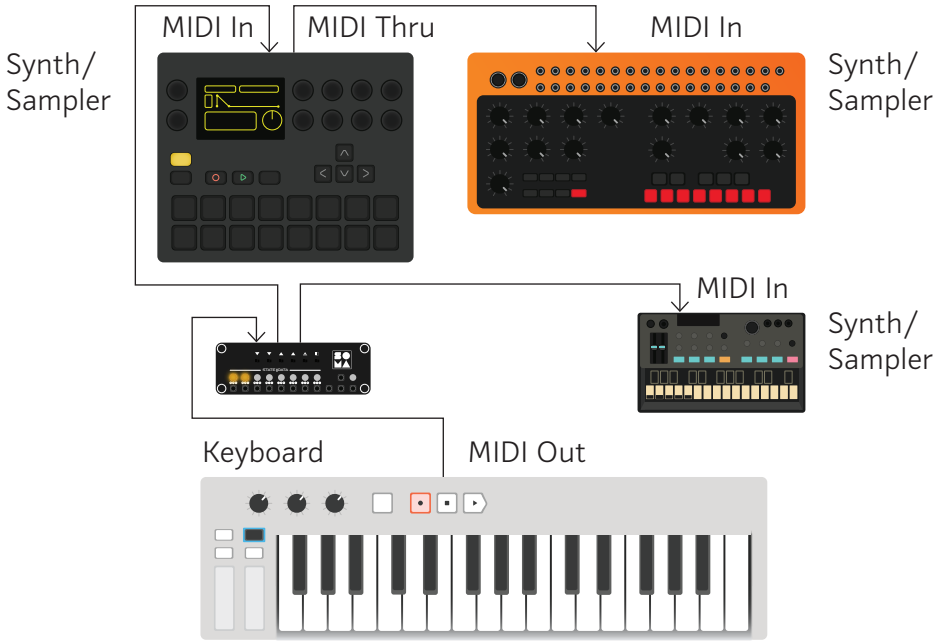
Sometimes you don't need to invent something radically new, it's enough to wisely redistribute the possibilities that already exist. Metaconformer does exactly this.

# COMBINER TUTORIAL

**Connection**

**1.** Connect the mini-USB cable to power Metaconformer

**2.** Connect the MIDI output of your MIDI keyboard to the MIDI input of the Metaconformer




Use only stereo mini-TRS to MIDI cables to connect to MIDI inputs and outputs. Using mono cables may damage the device.

Do not apply signals higher than 3.3V to MIDI inputs and outputs. Connect these jacks only to the MIDI inputs and outputs of other MIDI devices. When applying CV or other signals, a miracle will not happen, but there is a risk of damaging the device.

**3.** Connect the MIDI outputs of Metaconformer to the MIDI inputs of the instruments you want to play.

You can connect instruments to all four outputs of Metaconformer, or chain them together via MIDI thru.

The fourth MIDI output of Metaconformer is hybrid and can also be used as a sync output instead. For settings for this output, see the Hybrid output section.

**Communication CC knob**

After turning on Metaconformer, the operating mode must be selected. To do this, just turn any knob that sends a CC message. For example, the modulation wheel of a MIDI keyboard. Let's agree to call it the Communication CC, because with this wheel (or knob) you will communicate with Metaconformer and will be able to change its settings. You can assign the Communication CC function to a different knob any time Metaconformer is turned on or restarted. If available, using a modwheel is practical for this role.

The Communication knob allows you to configure Metaconformer when you are in the different menus. In this case, CC messages will be consumed by Metaconformer. Otherwise, CC messages are sent to the MIDI output unchanged.

When the Communication knob is selected, only the CC number of the message is stored, not its channel. So it doesn't matter which MIDI channel the CC communication message comes from. Metaconformer will respond to this CC message from any channel.

You can use the same CC for multiple Metaconformers. It is important that only one Metaconformer is in the Learn or Config menu.

STATE/DATA display

The data on the 8-bit display is displayed on orange LEDs, which can be in several states:
● — LED is on,
● — LED is off,
◑ — LED is blinking.

If in the figure one LED is in the ◑ state, and the other is in the ◑ state, then their on/off states are inverse at any time.

**Operating Mode**

Select the second-from-the-left LED on the State/data display to select Combiner Mode.

To launch Metaconformer with the settings of the previous session, just press the CFG button after power up. The same communication CC will be used and all settings will be loaded from memory.

**MIDI Thru**

In Combiner mode, Metaconformer lets all incoming MIDI notes and messages pass through it unchanged, except for those coming from the input channel.

The passage of each note is accompanied by a flashing green LED. One blink on noteOn and one on noteOff.

Messages from the input channel are sent to the processing unit of the Metaconformer and are sent to the output already converted, or are absorbed if no cell is enabled.

Go through all the relevant MIDI channels of the instruments you have connected to Metaconformer and make sure they hear notes. In case of problems, go to the Connection Test section.

### Notation of MIDI messages

The remainder of the tutorial will describe how the MIDI processor works, what happens to incoming messages, and how they are converted before being sent. All MIDI messages will be written as already decoded values of the status and data bytes. The status byte stores two values at once, which are separated by an underscore. For details, see MIDI Protocol.

For example, a keypress message
*noteOn_ch5 note50 vel50*
means that a note was pressed on channel 5, note number 50, velocity 50.
Message
*noteOff_ch5 note50 vel0*
means that the key with note 50 on channel 5 is released.

A message from a continuous controller (CC) looks like this
*CCmsg_ch7 param42 val100*
And means that the message stores a CC message on channel 7 with parameter number 42 and value 100.

### Learn Menu

To combine instruments into a hybrid polyphonic synthesizer, you need to set up the Metaconformer processor cells.

It is important to set up cells in complete MIDI silence, when there are no other messages on the channels, the sequencer is turned off, and all keys are released.

It is like a dialogue between user and Metaconformer. If there are other messages, Metaconformer may misunderstand your intent and set up cells incorrectly.

In Combiner mode, each Metaconformer cell is capable of storing the MIDI channel number that will become one of the voices. Setup is quick and easy.

**1.** On your MIDI keyboard, switch to the MIDI channel of the desired voice. Press any key to play a sound on the instrument. The green LED should flash rapidly each time a key is pressed and released.

Many samplers and drum machines only accept a limited range of notes on a MIDI channel, so you need to make sure that the instrument responds to your messages and plays sound.

**2.** Use the ◀ and ▶ buttons to select one of the eight Metaconformer cells.
**3.** Go to the Learn menu (LRN button). The green LED indicates this.
**4.** Play any note on this channel. The LEDs will display the MIDI channel number in an inverted binary representation, where a one in the digit corresponds to an off LED and a zero to an on LED. Learn more in the Binary Coded Numbers section.
**5.** Exit Learn menu by pressing LRN.

Ready! Metaconformer received the message
*noteOn_ch5 noteX velX* (note and velocity are not interesting, so marked with X)
and remembered the channel of the incoming note, thus setting up the cell. You should have heard the sound of the instrument that became a voice of the hybrid synthesizer. Setup completed.

In your case the values will be different.
Metaconformer sets up the cell according to the last received message

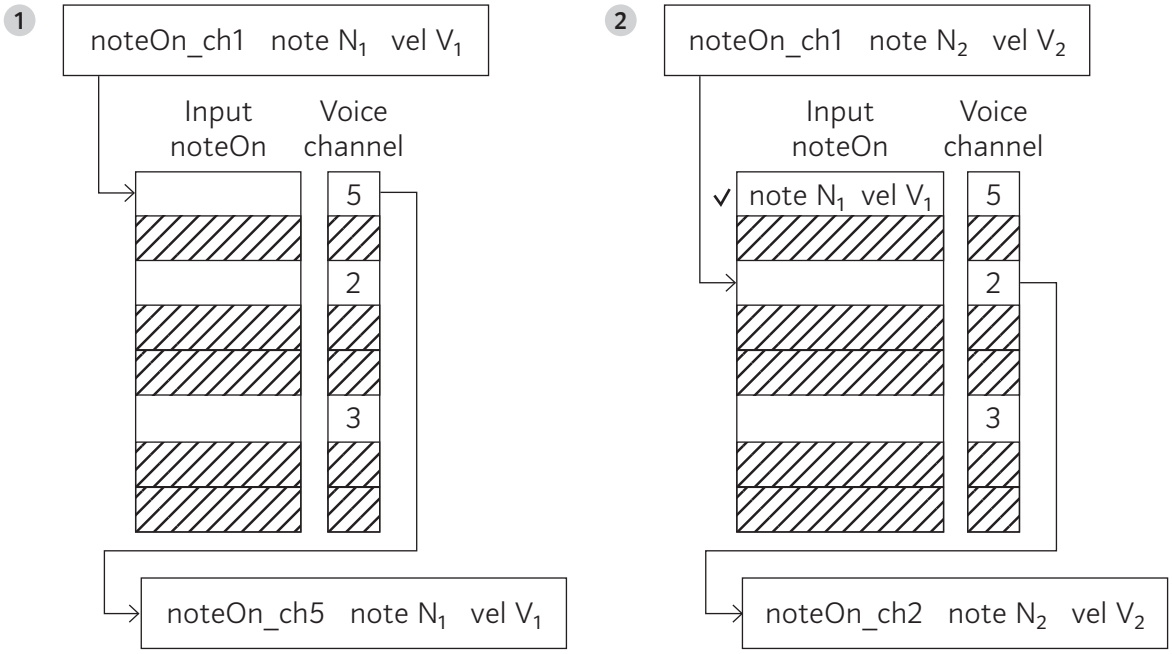
**Play Mode**

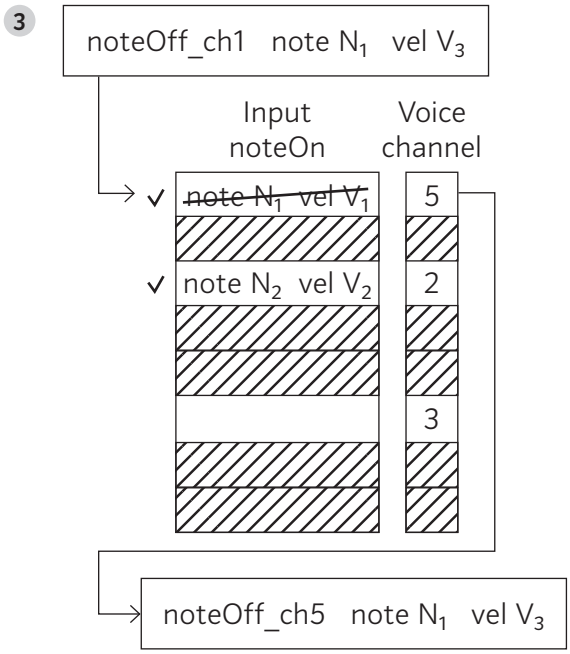
On your MIDI keyboard, switch to the input channel of Metaconformer. By default, this is channel 1. It'll be shown how to change this channel below.

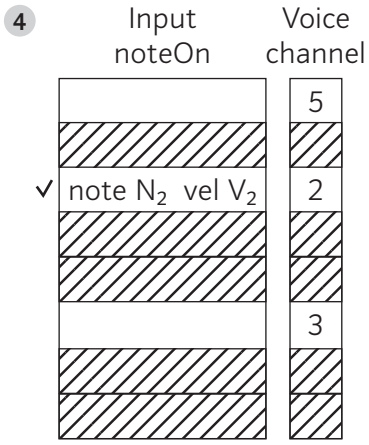
You can now play polyphonically on your new hybrid synthesizer!

When a noteOn message arrives on the input channel of Metaconformer, it starts looking for a free voice that is not yet occupied. The search proceeds sequentially through the eight cells, starting from the cell pointed to by the cursor.

An example with several different situations.

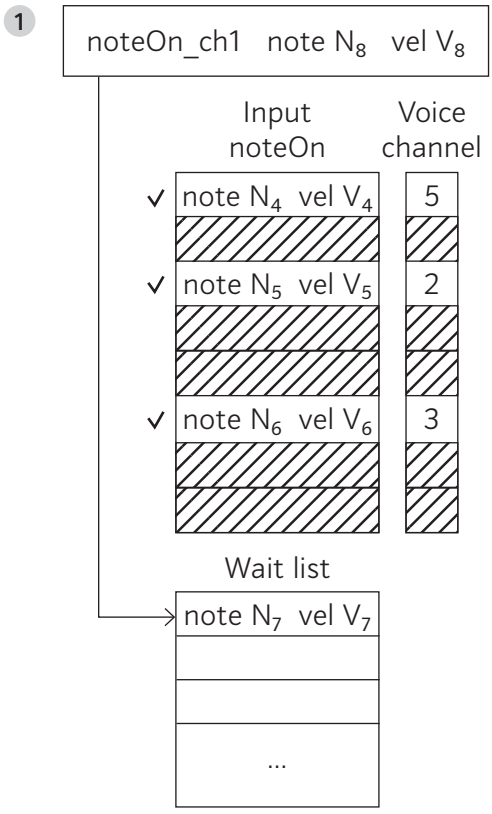


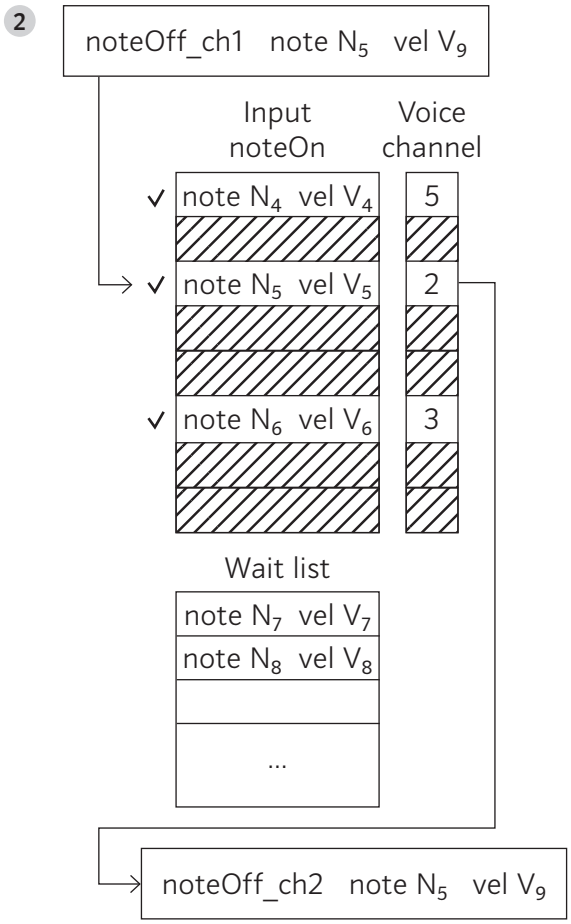


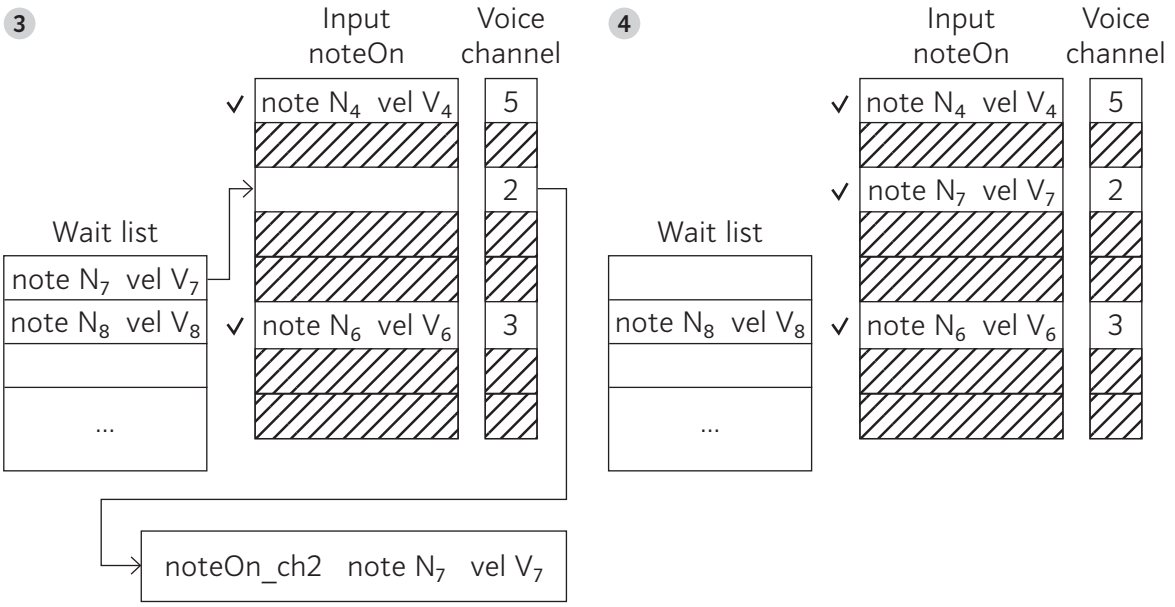




If there are no more free cells, and another noteOn message comes from the keyboard, Metaconformer saves it to the waiting list.

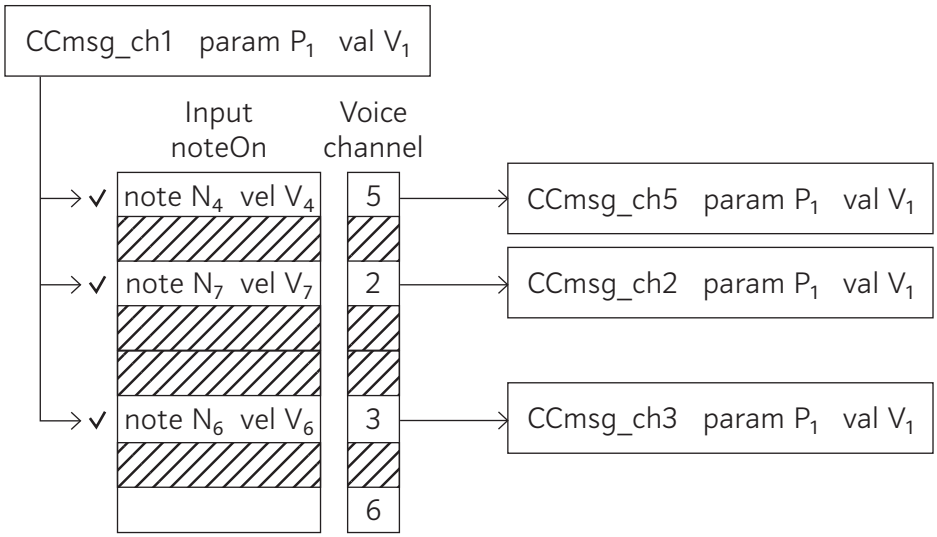
As soon as one of the cells is freed by an incoming noteOff message, Metaconformer checks the waiting list. If it is not empty, then a note from the waiting list is placed in this cell.






If CC, pitch bend, modulation wheel or channel aftertouch messages come from the input channel of the Metaconformer, then these messages are duplicated in the channels of all voices active at that moment.




This duplication of control signals is very useful when working with multi-timbral synthesizers and samplers. This allows the rotation of a single CC knob to configure several instruments at once.

Metaconformer searches for a free voice starting from the cell pointed to by the cursor. Move the cursor with the ◀ and ▶ buttons to change the sequence of activated voices.

The 8-bit STATE/DATA display, inspired by vintage computers, indicates by lighting the corresponding LEDs which voices are currently active.

Metaconformer allows you to simply mute previously added voices. Pressing the cell button temporarily disables the voice without deleting it.

**Delete**

Deleting a voice is as easy as setting it up.

**1.** Use the ◀ and ▶ buttons to select a cell
**2.** Go to Learn menu
**3.** Hold LRN to delete parameters
**4.** Release LRN to exit

**Input channel**

When you enter the Configuration menu by pressing the CFG button, you enter the Global Configuration menu of Metaconformer. The green LED starts blinking.

See the Feedback and General Function Tutorial for details on the Global Configuration menu. The instructions below will only show you how to configure the input channel of Metaconformer.

**1.** Use the ◀ and ▶ buttons to move the cursor to the eighth LED from the left, it should blink.
**2.** Use the Communication CC knob to set the desired channel number. The display shows binary coded numbers.

Input channel 5:
Input channel 14:

Ready! The input channel is set. When you exit the CFG menu, all settings are saved to memory automatically.

**Save**

The current cell configuration can be saved to memory. To do this, go to the Learn menu in any cell by moving the cursor to the cell and pressing LRN. Then exit it by pressing LRN while holding the CFG button.

# SPLITTER TUTORIAL

## Connection

**1.** Connect the USB-mini cable to power Metaconformer
**2.** Connect the MIDI output of your MIDI keyboard to the MIDI input of the Metaconformer

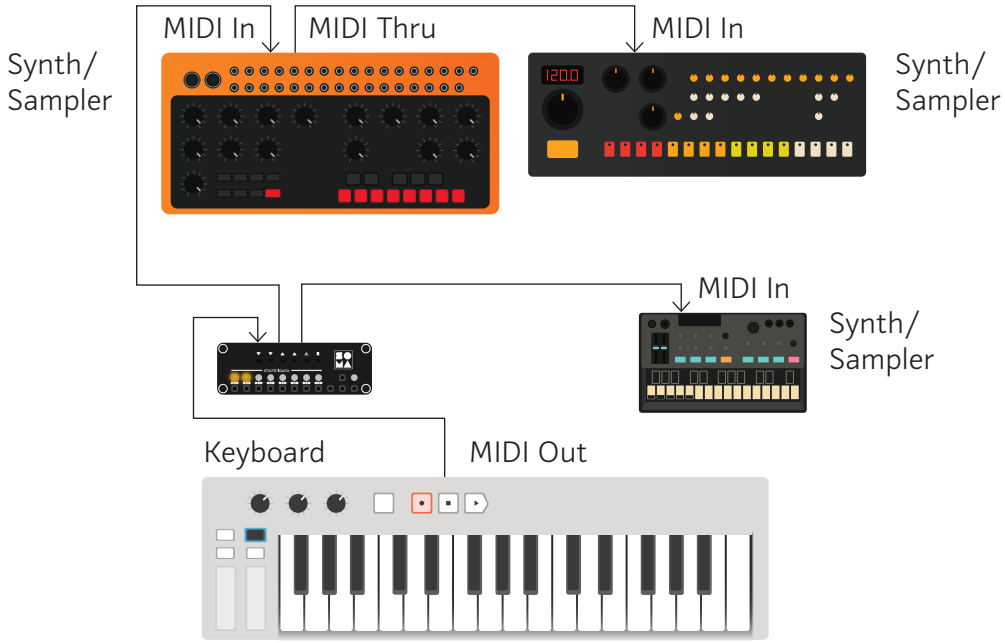
Use only stereo mini-TRS to MIDI cables to connect to MIDI inputs and outputs. Using mono cables may damage the device.

Do not apply signals higher than 3.3V to MIDI inputs and outputs. Connect these jacks only to the MIDI inputs and outputs of other devices. When applying CV or other signals, a miracle will not happen, but there is a risk of damaging the device.

**3.** Connect the MIDI outputs of Metaconformer to the MIDI inputs of the instruments you want to play.

You can connect instruments to all four outputs of Metaconformer, or chain them together via MIDI thru.

The fourth MIDI output of Metaconformer is hybrid and can also be used instead as a sync output. For settings for this output, see the Hybrid Output section




## Communication CC knob

After turning on Metaconformer, the operating mode must be selected. To do this, just turn any knob that sends a CC message. For example, the modulation wheel of a keyboard. Let's agree to call it the communication CC, because with this wheel (or knob) you will communicate with the Metaconformer and will be able to change its settings. You can assign the communication CC function to a different knob any time Metaconformer is turned on or restarted. If available, using a modwheel is practical for this role.

The Communication knob allows you to configure Metaconformer when you're in the different menus. In this case, CC messages will be consumed by the Metaconformer. Otherwise, CC messages are sent to the MIDI output unchanged.

When the Communication knob is selected, only the CC number of the message is stored, not its channel. So it doesn't matter which MIDI channel the CC communication message comes from. The Metaconformer will respond to this CC message from any channel.

You can use the same CC for multiple Metaconformers. It is important that only one Metaconformer is in the Learn or Config menu.

**STATE/DATA DISPLAY**

The data on the 8-bit display is displayed on orange LEDs, which can be in several states:
● — LED is on,
● — LED is off,
◑ — LED is blinking

If in the figure one LED is in the ◑, state, and the other is in the ◑, state, then their on/off states are inverse at any time.

**Operating Mode**

Select the first LED from the left on the State/data display to select Splitter Mode.

To launch Metaconformer with the settings of the previous session, just press the CFG button after power up. The same communication CC will be used and all settings will be loaded from memory.

**MIDI Thru**

In Splitter mode, Metaconformer lets all incoming MIDI notes and messages pass through it unchanged, except for those coming from the input channel.

The passage of each note is accompanied by a flashing green LED. One blink on noteOn and one on noteOff.

Messages from the input channel are sent to the processing unit of the Metaconformer and are sent to the output already converted, or are absorbed if no cell is enabled.

Go through all the MIDI channels of the instruments you have connected to Metaconformer and make sure they hear notes. In case of problems, go to the Connection Test section.

### Notation of MIDI messages

The remainder of the tutorial will describe how the MIDI processor works, what happens to incoming messages, and how they are converted before being sent. All MIDI messages will be written as already decoded values of the status and data bytes. The status byte stores two values at once, which are separated by an underscore. For details, see MIDI Protocol.

For example, a keypress message
*noteOn_ch5 note50 vel50*
means that a note was pressed on channel 5, note number 50, velocity 50.
Message
*noteOff_ch5 note50 vel0*
means that the key with note 50 on channel 5 is released.

A message from a continuous controller (CC) looks like this
*CCmsg_ch7 param42 val100*
And means that the message stores a CC message on channel 7 with parameter number 42 and value 100.

### Learn Menu

To play several instruments at once from a single MIDI channel, depending on the position of the note on the keyboard, you first need to set up the Metaconformer cells.

It is important to set up cells in complete MIDI silence, when there are no other messages on the channels, the sequencer is turned off, and all keys are released.

It is like a dialogue between user and Metaconformer. If there are other messages, Metaconformer may misunderstand your intent and set up cells incorrectly.

In Splitter mode, each Metaconformer cell is capable of storing:
- MIDI channel number that will become the tone assigned to the note range
- Start note number of the range
- End note number of the range
- One of 16 range transposition options

Cell setup is quick and easy.

**1.** On your MIDI keyboard, switch to the MIDI channel of the desired voice. Press any key to play a sound on the connected instrument. The green LED should flash rapidly each time a key is pressed and released.

Many samplers and drum machines only accept a limited range of notes on a MIDI channel, so you need to make sure that the instrument responds to your messages and plays sound.

**2.** Use the ◀ and ▶ buttons to select one of the eight Metaconformer cells
**3.** Go to the learn menu (LRN button). The green LED indicates this.
**4.** Press any two notes on this channel, which should become the boundaries of the range. The lower nibble of the LEDs will display the MIDI channel number in inverted binary representation, where a 1 in digit corresponds to an off LED and a 0 to a lit one. Learn more about the binary representation of numbers.

**5.** The high nibble displays 1 of 16 range transposition options in inverted binary representation. By default, this value is 8 or 0001 on the Metaconformer LEDs. Another value can be selected by turning the communication knob. For more details, see the transposition setting section.
**6.** Exit learn menu

Ready! The Metaconformer receive save the channel of incoming notes, determined the boundaries of the range, and set up the cell itself. Setup completed. Fast and easy.

Ready! Metaconformer received the messages
noteOn_ch5 noteN1 velX (note and velocity are not interesting, so marked with X)
noteOn_ch5 noteN2 velX (note and velocity are not interesting, so marked with X)
and remembered note boundary and also the channel of the incoming notes, thus setting up the cell . You should have heard the sound of the instrument that became a range of the hybrid synthesizer . Setup completed .

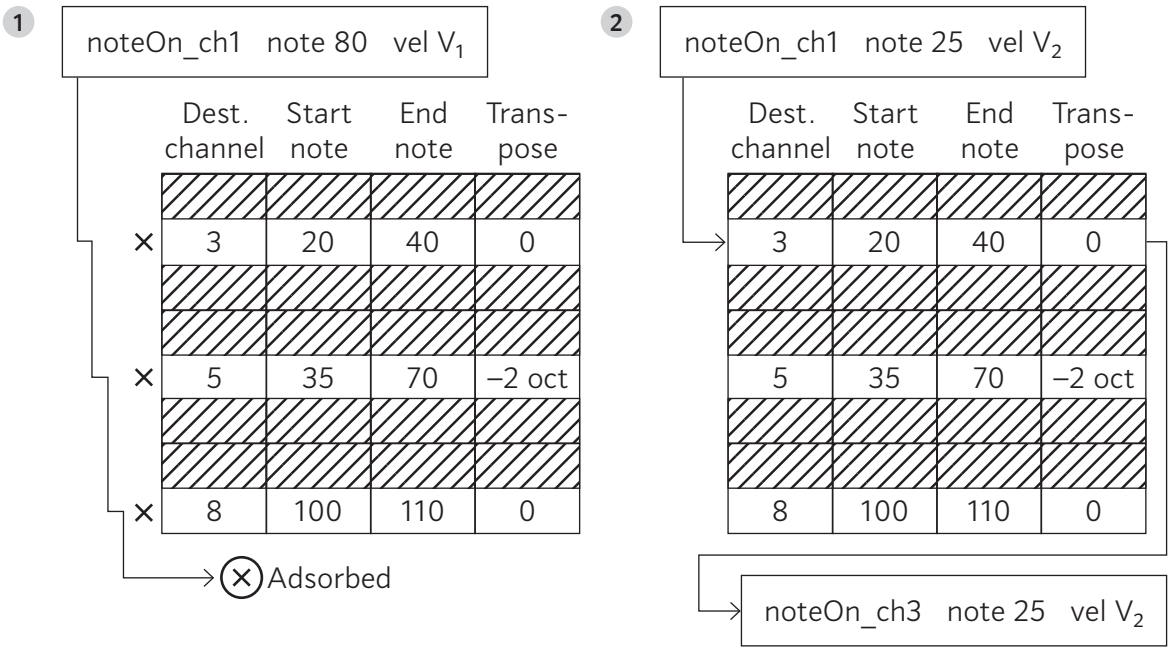
In your case the values will be different.
Metaconformer adjusts the cell according to the last incoming message.

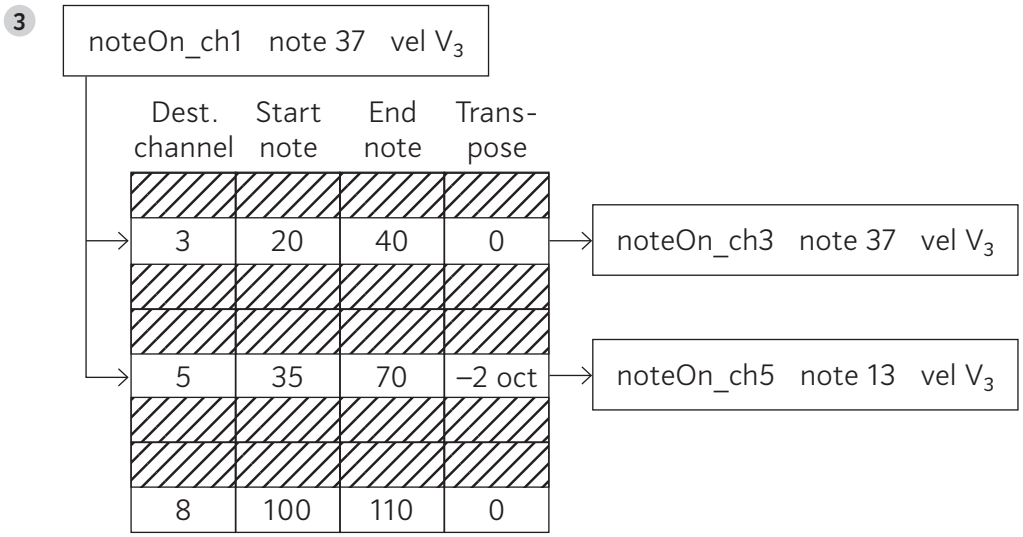
**Play Mode**

On your MIDI keyboard, switch to the input channel of Metaconformer. By default, this is MIDI channel 1. It'll be shown how to change this channel below.

You can now play multiple instruments on the same MIDI channel.

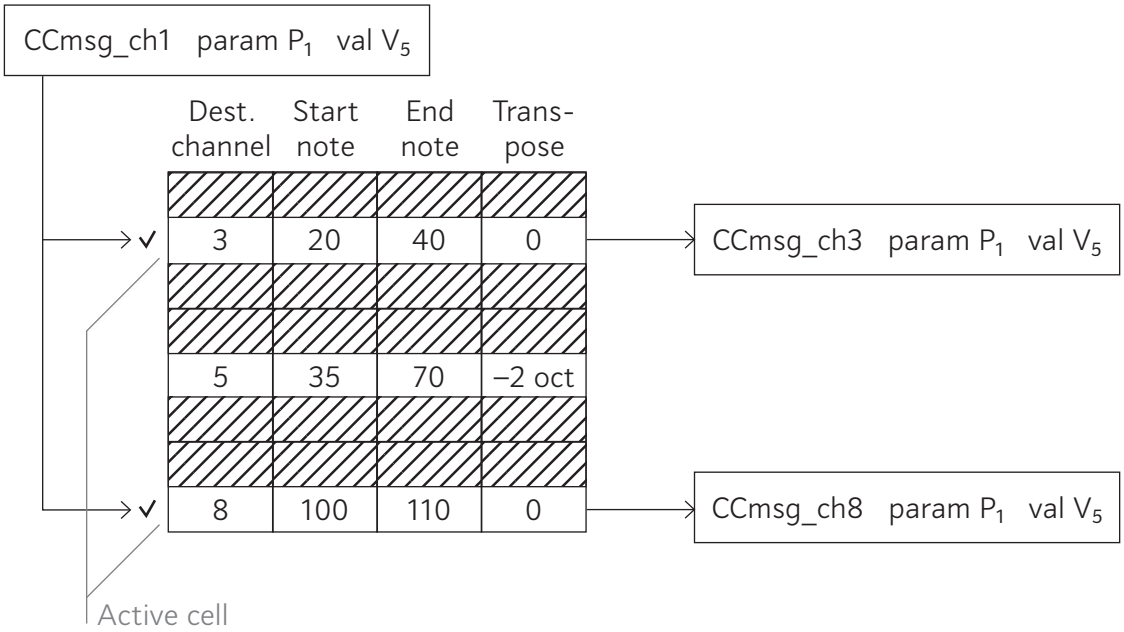
When a noteOn or noteOff message comes from the input channel of Metaconformer, it checks which enabled ranges contain the received note and sends it to that instrument, changing the value of the note taking into account the transposition setting.






Note ranges can overlap. If a note matches multiple ranges, it sends to all those ranges.

If CC, pitch bend, modulation wheel or channel aftertouch messages come from the input channel of Metaconformer, then these messages are duplicated to the channels of all ranges active at that moment.




This duplication of control signals is very useful when working with multi-timbral synthesizers and samplers. This allows the rotation of a single CC knob to configure several instruments at once.

The 8-bit STATE/DATA display, inspired by vintage computers, indicates which ranges are currently active.

Metaconformer allows you to simply mute previously added ranges. Pressing on the cell button allows you to temporarily disable the range without deleting it.

### Transposition

Metaconformer allows you to change the note value before sending it to the instrument. To do this, in each cell, you can choose 1 of 16 transposition options.

Go to the Learn menu and use the Communication knob to select one of the values that are set by a binary number, where 1 corresponds to the off LED, and 0 to the on LED.

0 — ●●●● — merge range into bottom not
1 — ●●●● — transposition –1 octave
2 — ●●●● — transposition –2 octave
3 — ●●●● — transposition –3 octave
4 — ●●●● — transposition –4 octave
5 — ●●●● — transposition –5 octave
6 — ●●●● — extension from top note down to 0
7 — ●●●● — merge range into high note
8 — ●●●● — no transposition
9 — ●●●● — +1 octave transposition
10 — ●●●● — +2 octave transposition
11 — ●●●● — +3 octave transposition
12 — ●●●● — +4 octave transposition
13 — ●●●● — +5 octave transposition
14 — ●●●● — extension from bottom note up to 127
15 — ●●●● — range inversion

You can create two identical note ranges corresponding to different instruments. Without transposition, the timbres will simply merge with each other. Transposing one range in relation to another will give an interesting octave layering of sounds.

Transposition with range extension is designed specifically to work in conjunction with Metaconformer in Translator mode, which translates incoming notes into CC messages that change instrument parameters.
You can create a small range of keys, which, after passing through the Metaconformer Translator, turns into coarse parameter keys. You can set several of these ranges to quickly adjust the sound without having to turn any knobs.

### Delete

Deleting a note range is as easy as setting it up.
**1.** Use the ◀ and ▶ buttons to select a cell
**2.** Go to Learn menu
**3.** Hold LRN to delete parameters
**4.** Release LRN to exit

### Input channel

When pressing the CFG button, you enter the Global Configuration menu of Metaconformer. The green LED starts blinking.

See the Feedback and General Function Tutorial for details on the Global Configuration menu. The instructions below will only show you how to configure the input channel of Metaconformer.

**1.** Use the ◀ and ▶ buttons to move the cursor to the eighth LED from the left, it should blink.
**2.** Use the Communication CC knob to set the desired channel number. The screen shows binary coded numbers.

Input channel 5:
Input channel 14:

Ready! The input channel is set. When you exit the CFG menu, all settings are saved to memory automatically.

### Save

The current cell configuration can be saved to memory. To do this, go to the Learn menu in any cell by moving the cursor to the cell and pressing LRN. Then exit it by pressing LRN while holding the CFG button.

# TRANSLATOR TUTORIAL

**Connection**

**1.** Connect the mini-USB cable to power Metaconformer
**2.** Connect the MIDI output of your MIDI keyboard to the MIDI input of the Metaconformer

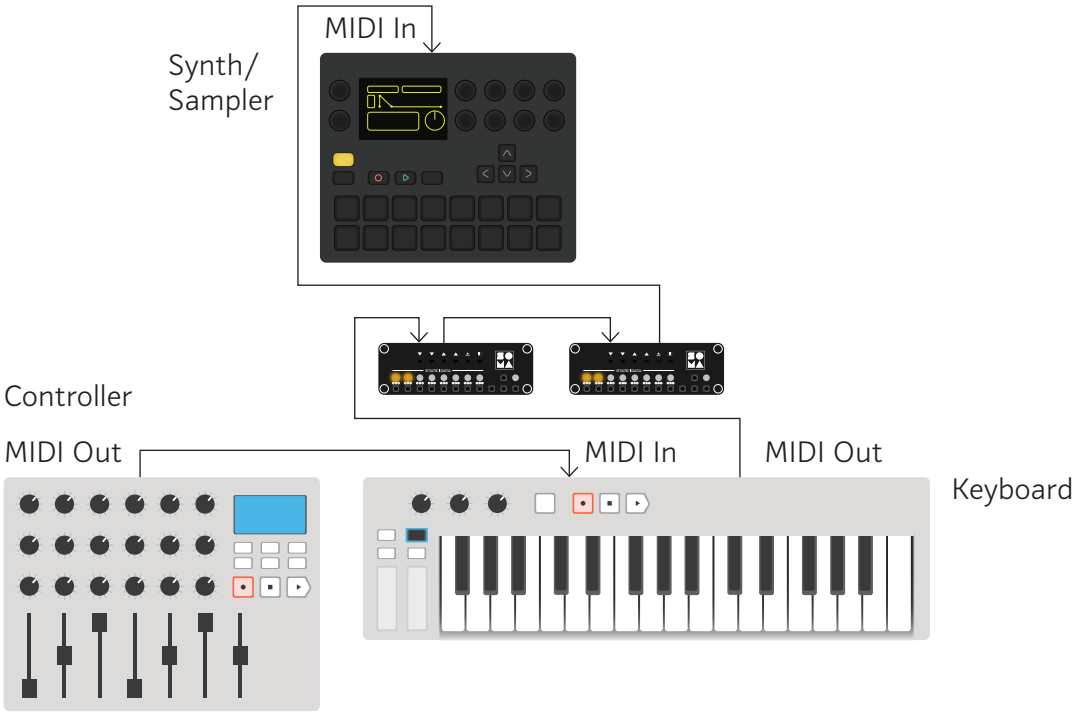
Use only stereo mini-TRS to MIDI cables to connect to MIDI inputs and outputs. Using mono cables may damage the device.

Do not apply signals higher than 3.3V to MIDI inputs and outputs. Connect these jacks only to the MIDI inputs and outputs of other devices. When applying CV or other signals, a miracle will not happen, but there is a risk of damaging the device.

**3.** Connect the MIDI outputs of Metaconformer to the MIDI inputs of the instruments you want to play.

You can connect instruments to all four outputs of Metaconformer, or chain them together via MIDI thru.

The fourth MIDI output of Metaconformer is hybrid and can also be used as a sync output instead. For settings for this output, see the Hybrid Output section.




**Communication CC knob**

After turning on Metaconformer, the operating mode must be selected. To do this, just turn any knob that sends a CC message. For example, the modulation wheel of a keyboard. Let's agree to call it the Communication CC, because with this wheel (or knob) you will communicate with the Metaconformer and will be able to change its settings. You can assign the Communication CC function to a different knob any time Metaconformer is turned on or restarted. If available, using a modwheel is practical for this role.

The Communication knob allows you to configure Metaconformer when you are in the different menus. In this case, CC messages will be consumed by the Metaconformer. Otherwise, CC messages are sent to the MIDI output unchanged.

When the Communication knob is selected, only the CC number of the message is stored, not its channel. So it doesn't matter which MIDI channel the CC communication message comes from. Metaconformer will respond to this CC message from any channel.

You can use the same CC for multiple Metaconformers. It is important that only one Metaconformer is in the Learn or Config menu.

### STATE/DATA DISPLAY

The data on the 8-bit display is displayed on orange LEDs, which can be in several states:
● — LED is on,
● — LED is off,
◑ — LED is blinking.

If in the figure one LED is in the ◑ state, and the other is in the ◑ state, then their on/off states are inverse at any time.

### Operating Mode

Select the third-from-the-left LED on the state/data display to select Translator Mode.

●●●●●●●

To launch Metaconformer with the settings of the previous session, just press the CFG button after power up. The same communication CC will be used and all settings will be loaded from memory.

### MIDI Thru

In Translator mode, Metaconformer lets all incoming MIDI notes and messages pass through it unchanged, except for those coming from input channels of cells that are currently configured, enabled, and active.

The passage of each note is accompanied by a flashing green LED. One blink on noteOn and one on noteOff.

Messages from the input channels of the cells are sent to the processing unit of Metaconformer and sent to the output already converted.

Go through all the MIDI channels of the instruments you have connected to Metaconformer and make sure they hear notes. In case of problems, go to the Connection Test section

### Notation of MIDI messages

The remainder of the tutorial will describe how the MIDI processor works, what happens to incoming messages, and how they are converted before being sent. All MIDI messages will be written as already decoded values of the status and data bytes. The status byte stores two values at once, which are separated by an underscore. For details, see MIDI Protocol.

For example, a keypress message
*noteOn_ch5 note50 vel50*
means that a note was pressed on channel 5, note number 50, velocity 50.

Message
*noteOff_ch5 note50 vel0*
means that the key with note 50 on channel 5 is released.

A message from a continuous controller (CC) looks like this
*CCmsg_ch7 param42 val100*
and means that the message stores a CC message on channel 7 with parameter number 42 and value 100.

### Learn Menu

To translate notes coming to the input of Metaconformer cells into CC messages to control the sound parameters, you first need to set up the Metaconformer cells.

It is important there is complete MIDI silence when you set up cells, with no other messages on the channels, the sequencer is turned off, and all keys are released.

It is like a dialogue between user and Metaconformer. If there are other messages, Metaconformer may misunderstand your intent and set up cells incorrectly.

In this mode, each Metaconformer cell is capable of storing:
**1.** CC number corresponding to the controlled parameter
**2.** MIDI channel number whose notes will be translated into the CC value of the knob
**3.** MIDI channel number of the voice whose parameter we will control

Setup is quick and easy. However, first it is necessary to find out from the user manual of the connected instrument which CC number corresponds to the desired parameter.

If you use a MIDI controller, set one of its knobs to control this particular parameter of the instrument. To do this, you need to specify the CC and MIDI channel of the instrument in the controller menu. With the preparation complete, you can proceed to setting up the Metaconformer cell.

**1.** On your MIDI keyboard, you need to switch to the MIDI channel, the notes of which we are going to translate into CC values.
**2.** Use the ◀ and ▶ buttons to select one of the eight Metaconformer cells.
**3.** Go to the Learn menu (LRN button). The green LED indicates it's active.
**4.** Turn the knob on the MIDI controller to send messages to the Metaconformer CC that change the desired parameter of the instrument.
**5.** Press any key on the MIDI keyboard.
**6.** Exit the Learn menu.

Ready! Metaconformer received message
*CCmsg_ch7 param23 valX* (parameter value is not interesting, therefore X is marked)
memorized the channel of the incoming CC and the number of this CC,
got a message
*noteOn_ch6 noteX velX* (note and velocities are not interesting, so marked with X)
remembered the channel of the note, and automatically adjusted the cell. Setup completed.
Fast and easy.

In your case the values will be different.
Metaconformer adjusts the cell according to the last incoming message.

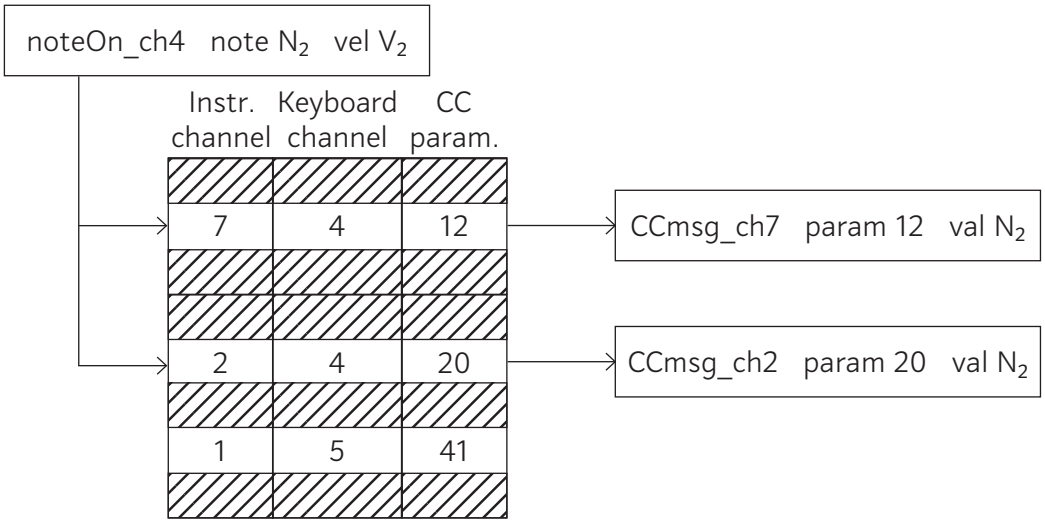
You can set up a Translator cell with the Communication knob.

**1.** Use the ◀ and ▶ buttons to select one of the eight Metaconformer cells.
**2.** Enter the Learn menu (LRN button). The green LED indicates it's active.
**3.** Use the ◀ and ▶ buttons to position the flashing cursor on the first LED from the left.
**4.** Use the Communication knob to set the desired CC value. The values are set in inverted form, where the 1 corresponds to the off LED, and 0 to the on LED.
CC parameter 17:
**5.** Use the ◀ and ▶ buttons to position the flashing cursor on the second LED from the left.
**6.** Use the Communication knob to set the desired channel from which notes will be translated into CC messages.
keyboard channel 5:
**7.** Use the ◀ and ▶ buttons to position the flashing cursor on the third LED from the left.
**8.** Use the Communication knob to set the channel value of the instrument whose CC parameter you want to change from the keyboard.
keyboard channel 7:

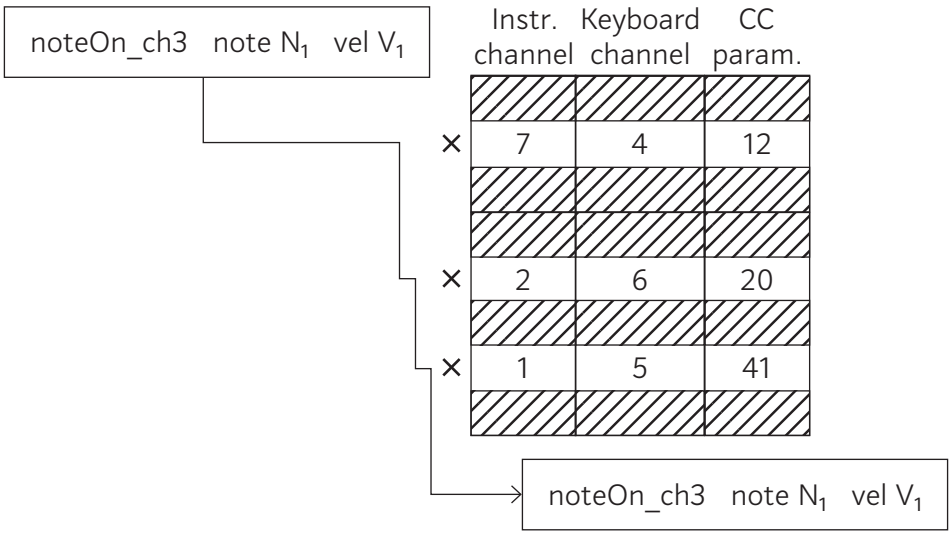
### Play mode

On your MIDI keyboard, switch to the cell's input channel, the channel whose notes will be translated in CC messages.

When the noteOn message arrives to Metaconformer, it checks in which enabled cells of the translator the input channel corresponds to this note, and sends a CC message to the instrument instead of the CC note, and absorbs the note.

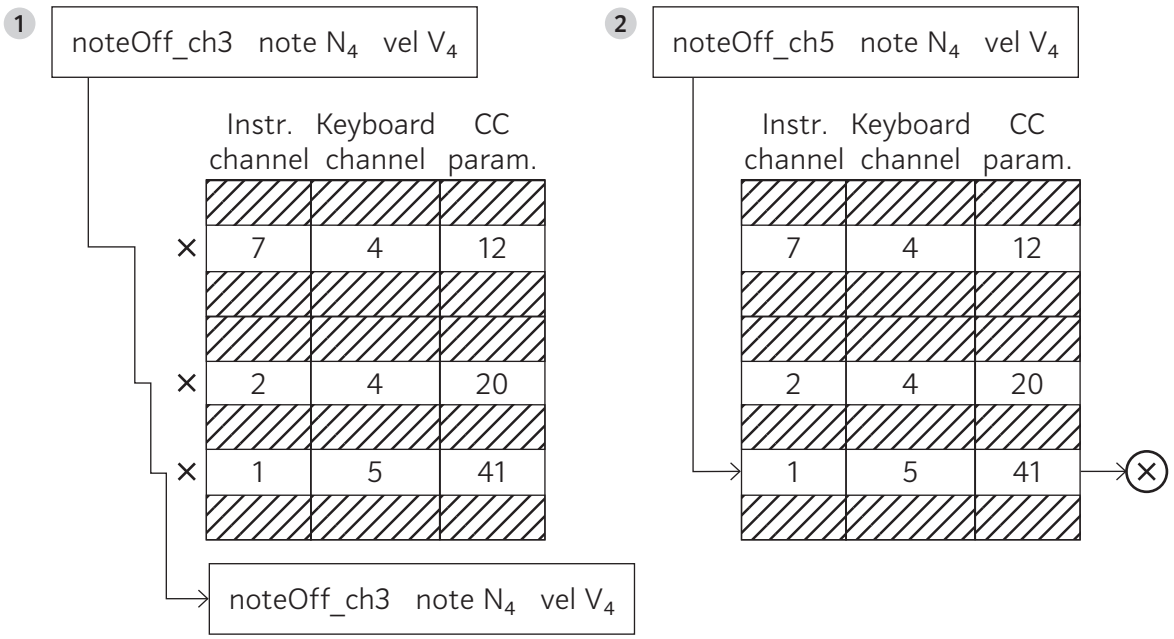



For example, the instrument is listening to channel 5. This means that its VCA is triggered by notes from channel 5, and the parameters are controlled by CC messages from channel 5.
If we translate notes into CC messages from the same channel 5, then we will have nowhere to get the notes to activate the VCA, because the Metaconformer absorbs the note that arrived to the translator. Therefore, we will get silence, in which the value of one of the parameters changes.
So, you need to take notes from another MIDI channel. For example, channel 4. In this case, when receiving a note from channel 4, the Metaconformer will take its value from this note and turn it into the value of the parameter on channel 5.



If the note does not correspond to any enabled cell of the translator, then the message is sent to the output unchanged.

When a noteOff message is received by Metaconformer, it checks in which enabled cells of the translator the input channel matches that note. If the note does not correspond to any cell of the translator, then the message is sent to the output unchanged.




Otherwise, it is absorbed by Metaconformer.

Other types of messages pass through Metaconformer unchanged.

The 8-bit display indicates by flashing the cell when it is translating a note into a parameter value.

Metaconformer allows you to simply mute a previously added Translator. Pressing the cell button will mute the enabled cell.

If this Translator cell is muted, the notes will no longer be processed, but will simply be sent to the output.

### Using multiple Metaconformers

The main disadvantage of Metaconformer's Translator mode is that two keyboards are needed. One MIDI channel activates the VCA envelope, and the second channel sends notes that are translated into values.

The solution is to use a second Metaconformer in Combiner mode.

**1.** Connect the second Metaconformer between the MIDI controller and Metaconformer in Translator mode.
**2.** Run it in Combiner mode.
**3.** Set cell 1 to the channel that activates VCA.
**4.** Set cell 2 to the channel that sends notes.
**5.** Set your MIDI keyboard to the input channel of the Metaconformer in Combiner mode.

Now you can very conveniently use a single MIDI channel for both triggering the VCA, and also for changing parameters of the instrument.

Similarly, you can use Metaconformer in Splitter mode.

### Delete

Deleting a voice is as easy as setting it up.
**1.** Use the ◀ and ▶ buttons to select a cell.
**2.** Go to Learn menu
**3.** Hold LRN to delete parameters
**4.** Release LRN to exit

### Save

The current cell configuration can be saved to memory. To do this, go to the Learn menu in any cell by moving the cursor to the cell and pressing LRN. Then exit it by pressing LRN while holding the CFG button.

# FEEDBACK AND GENERAL FUNCTION TUTORIAL

This tutorial contains all information about the various functions of Metaconformer outside of MIDI processing, and also describes in detail how the feedback works.

## Hardware

Metaconformer has two MIDI inputs, messages from which are combined into a single stream.

Metaconformer has three MIDI outputs.

The fourth output of Metaconformer is hybrid and can operate in two modes: MIDI output or sync signal generator.

All MIDI inputs and outputs have an A and B type switch that can act as a hardware on/off switch.

To avoid damage to the device when working with MIDI outputs and inputs, use only stereo mini-jack TRS to MIDI cables.

## STATE/DATA DISPLAY

The Metaconformer provides the main feedback in the form of sounds. However, it also has a STATE/DATA display of 8 orange LEDs and one green LED. Despite its primitiveness, such an LED indicator provides the user with maximum necessary information.

The data on the 8-bit display is displayed on orange LEDs, which can be in several states:

The data on the 8-bit display is displayed on orange LEDs, which can be in several states:
● — LED is on,
● — LED is off,
◑ — LED is blinking.

If in the figure one LED is in the ◑ state, and the other is in the ◑ state, then their on/off states are inverse at any time.

## Play mode

As long as the green LED is permanently off, Metaconformer is in Play mode.

In Play mode, the green LED flashes quickly when MIDI thru notes pass through Metaconformer.

If no buttons are pressed, the state/data display shows active cells.

When the ◀ and ▶ buttons are pressed, the display shows a blinking cursor corresponding to the selected cell. As soon as the user has chosen a cell, Metaconformer returns to displaying active cells.

When cells are muted by pressing the button below the cell's LED, the STATE/DATA display shows the mute state of the remaining enabled cells.

**Learn menu**

By pressing the LRN button, Metaconformer enters the Learn menu of the selected cell.

To avoid errors in the MIDI stream, Metaconformer allows access to the Learn menu only when all cells are inactive.

Access the Learn menu by pressing the LRN button.

While Metaconformer is in the Learn menu, the green LED is on continuously.

In this menu, the display shows a minimum of information about the configuration of the selected Metaconformer cell in the format of binary coded numbers. Please note that it functions as a negative display - a 1 in the digit corresponds to an off LED ●, and a on LED ● corresponds to 0.

In Combiner mode, the cell shows the MIDI channel number of the voice in binary coded representation, with the least significant digit to the left.

Channel 1, the display shows ●●●●●●●●
Channel 5, the display shows ●●●●●●●●

In Splitter mode, the display shows the range MIDI channel number on the low nibble (4 orange LEDs on the left) and one of the 16 transposition types on the high nibble (4 orange LEDs on the right).

Channel 5, transposition -1 octave, the display shows ●●●●●●●●
Channel 3, transposition +4 octaves, the display shows ●●●●●●●●

When you press the ◀ and ▶ buttons, you will hear the lowest and the highest notes of the range on the selected instrument, taking into account the selected type of transposition of the range.

In Translator mode, the display shows a blinking cursor zone and the value of the corresponding parameter in the negative display is not blinking.

The cursor is moved with the ◀ and ▶ buttons. The parameter selected by the cursor can be changed with the Communication knob.

The cursor in the first position from the left indicates the CC number. The remaining 7 bits encode the CC number itself.
CC parameter 17: ◐●●●●●●●

Note that only the first LED on the left is blinking.

The cursor in the second position indicates the number of the MIDI channel being translated. That is, the channel, the notes from which will turn into the values of the synthesis parameters. The high nibble encodes the channel number itself.
keyboard channel 5: ◑◑◑◑●●●●

Note that since there are only 16 values, only the 4 most significant bits are required to display them. Therefore, the cursor area occupies the least significant 4 bits.

The cursor in the third position from the left indicates the channel number of the synthesizer. This channel listens to the synthesizer whose parameter you want to change by pressing notes. The high nibble encodes the channel number itself.
instrument channel 7:

**Global Configuration Menu**

Back to Play mode. The green LED is off.

To avoid errors in the MIDI stream, Metaconformer allows access to the Configuration menu only when all cells are inactive.

Switch to the Global Configuration menu by pressing the CFG button.

In this menu, the green LED is blinking. The STATE/DATA display shows a blinking cursor corresponding to the selected parameter, and the value of this parameter itself, which is in the range from 0 to 15. If the cursor is in the lower nibble (4 LEDs on the left), then the value is shown in the higher nibble (4 LEDs on the right). If the cursor is in the high nibble, then the value is indicated in the low nibble.

**Hybrid Output**

The fourth output of Metaconformer is hybrid and can work either as a MIDI output or as a sync signal source.

To set it up to work in MIDI output mode, you need to go to the Global Config menu by pressing CFG, and move the cursor to the first position from the left. Then set the value to 1 with the communication knob.

To set it up to work in sync signal mode, set the value of this parameter to 0 using the Communication knob.

By default, the sync signal has a positive polarity and a divider of 6.

A MIDI clock message is sent 24 times per 1/4 beat. So a divider of 6 provides a sync signal four times for every 1/4 beat, i.e. 1/16.

To change the polarity, place the cursor on the second LED from the left and set the value to 0 for positive polarity and 1 for inverted. The value 2 disables the sync signal.

Positive sync polarity:
Inverted sync polarity:
Sync off:

To change the divider, place the cursor on the third LED from the left and set the value from 2 to 16. Since 24 clocks per quarter note are transmitted in MIDI, the most appropriate values are 3, 6 and 12, since when using a sync signal, receiving devices expect a metronome at 1/8 or 1/16 beat. Using the rest of the dividers can give interesting rhythmic effects.

Divider 3:
Divider 6:
Divider 12:
Divider 15:
Divider 16:

Set up your sequencer to send MIDI clock to Metaconformer. To start Metaconformer's sync signal, it must receive a MIDI Start message. To turn off sync, a MIDI Stop message is required.

**Input Channel**

The eighth position of the cursor is always the input channel in Splitter and Combiner mode. This parameter is not used in Translator mode.

Input channel 5:
Input channel 14:

The remaining parameters are not reserved yet, but this may change in future firmware.

When you exit Global Configuration mode, all settings are stored in memory for the current operating mode.

The global settings of another mode may differ.

**Firmware Update**

To perform a firmware update, please do the following:
**1.** Connect Metaconformer to the computer via USB.
**2.** While the CFG button is held, press the RST button. Metaconformer will start in flash drive mode and will be detected as such on your computer, and all its LEDs will turn on.
**3.** Download the firmware from the SOMA website and copy it to Metaconformer.
**4.** Rename the file to be called blink.bin
**5.** Restart Metaconformer by pressing RST. It will start already flashed.
**6.** You can now disconnect Metaconformer from your computer.

Updating the firmware is a process that should be done carefully. We also do not recommend flashing the device with unverified firmware, as the result can be unpredictable and lead to the loss of all the patches you created.

# EXTRAS

This section contains additional information that may be useful during the operation of the Metaconformer.

**Connection Testing and Troubleshooting**

Immediately after connecting your MIDI keyboard, Metaconformer and instruments, you need to check the connection to rule out possible hardware problems.

Go to the relevant MIDI channels on your MIDI keyboard and make sure that the instruments are configured to the desired MIDI channels and are properly receiving outgoing messages from the keyboard.

Samplers and drum machines can typically only receive a limited range of MIDI notes specified in their instructions.

Metaconformer flashes a green LED each time it receives noteOn and noteOff messages, i.e. when a key is pressed and released, if it is in Play mode.

If it doesn't flash when pressed, you may be on a Metaconformer input channel. Change to a different MIDI channel on the keyboard and try sending notes from it.

If still no MIDI messages are registering, there may be a problem with the MIDI input of Metaconformer. Try plugging your keyboard into a different MIDI input of Metaconformer and follow the previous steps.

If that helps, then the first MIDI input is not functioning correctly. Contact technical support at somasynths@gmail.com for assistance.

If that still doesn't work, then either both MIDI inputs are bad, or there's a problem with the keyboard. Connect the keyboard directly to the instrument.

If the instrument still does not receive messages, then the problem is either in the keyboard or in the instrument. Make sure the instrument's MIDI input channel matches the selected channel on the keyboard. Also make sure the instrument is set to receive messages from its MIDI input.

If the instrument is now receiving messages, then there is a problem with both Metaconformer inputs. Contact technical support at somasynths@gmail.com for assistance.

**Binary coded numbers**

Metaconformer has a minimum of situations where you need to interact with binary numbers.

**1.** Input channel setup in Splitter and Combiner configuration menus
**2.** Setting the prescaler for the sync signal
**3.** Setting the transposition in the Splitter

In all these situations, you need to select one of the 16 options, which corresponds to a specific configuration of lit LEDs. Please refer to the table below, where binary numbers in a certain sense are just pictograms.

| | Learn | CFG |
|---|---|---|
| 0 | ○○○○ | ●●●● |
| 1 | ●○○○ | ○●●● |
| 2 | ○●○○ | ●○●● |
| 3 | ●●○○ | ○○●● |
| 4 | ○●○● | ○●○● |
| 5 | ●○●○ | ○●○● |
| 6 | ○●●○ | ●○○● |
| 7 | ●●●○ | ○○○● |
| 8 | ○○○● | ●●●○ |
| 9 | ●○○● | ○●●○ |
| 10 | ○●○● | ●○●○ |
| 11 | ●●○● | ○○●○ |
| 12 | ●●○○ | ○○●● |
| 13 | ●○●● | ○●○○ |
| 14 | ○●●● | ●○○○ |
| 15 | ●●●● | ○○○○ |

In other situations, the LEDs work as on/off indicators.

In our opinion, this solution is better than the addition of four symbolic indicators, which would increase the cost and also significantly burden the minimalistic appearance of the device.

While they may seem abstract at first sight, we will try to convince you that binary numbers are not fraught with something terrible, complex or magical.

To begin with, this is just a record of a number for the exchange of information. This entry is made in one of the many possible number systems. And number systems are just a tool that works better in some situations and worse in others.

Human civilization has been using the 10-digit number system based on the number of fingers (or digits) on our hands for a very long time.

However, there are alternative systems. For example, ancient merchants used the 12-digit number system based on the number of bones in the index, middle, little and ring fingers. This system made it easier to divide numbers, since 12 is divisible by 2,3,4,6. This system offers more options than just the 2 and the 5 in the 10-digit number system.

Another alternative system is the unary number system. For example, eight apples are 🍎🍎🍎🍎🍎🍎🍎🍎. Or 11111111 apples. In this system, the presence of something is indicated by writing the number 1 (essentially a notch) as many times as there is this something.

You have likely seen in movies how prisoners put notches on the wall of their cell to measure the number of days in captivity. Six notches, and the seventh is crossed out to mark the end of the week ~~111111~~. This is also a peculiar example of a unary number system, because it is not known in advance what number to write down.

If you cross out not every six marks, but every nine, then the number of 10s can be measured by the number of horizontal notches. The main idea behind the strikethrough is that you do not have to recalculate all the marks each time, because during the longer recalculation you are more likely to make mistakes, and the final result can be inaccurate. This is a kind of information caching. When you want to know the final number, you simply find out the number of tens and add uncrossed marks to it, that is, the number of ones.

But on the wall, you can select a special place where a different mark will be created, denoting tens. And when there are nine 10s, it will be possible to cross them out and get the number of tens of tens, that is, hundreds.

We have come to understand the idea of the digits of a number.
8 ones,
followed by 9 ones,
followed by 0 ones and 1 ten,
followed by 1 one and 1 ten, ...,
followed by 9 ones and 9 tens,
followed by 0 ones and 0 tens and 1 hundred,
followed by ....

Each time a digit goes from 9 to 0, the next digit increases by 1. But if it also goes from 9 to 0 at that moment, then the next digit after it should also increase by 1, and so on in the chain.

Since arithmetic originated in the ancient Arab countries, we write the digits not from left to right, as in the example above, but from right to left. 123 - 1 hundred, 2 tens and 3 units. Although if we agreed otherwise, it could well mean 3 hundreds, 2 tens and 1 unit.

Once again we want to emphasize that numbers are just an agreement of people about the way information is transmitted.

Computers are made up of logic gates that can either store a charge or not store it. Two states, 0 and 1. There have been attempts to make computers based on elements that can store three states. But these attempts were not successful. Two-state systems are incredibly scalable and reliable. Therefore, in computers, all numbers are stored in the binary system. And people who interact with these numbers operate with them quite confidently.

Let's calculate how much the unary number 1111111 will be in the binary number system. We will count in the same way as in the example with the prisoner, only we will cross out (increase the next digit) when 1 goes to 0, and not 9 to 0, as we are all used to.

We will take turns dragging one mark or serif from the left to the right side.
1111111 — transferred 0 — 0 units
111111 — moved 1 — 1 unit
11111 — moved 2 — 0 units 1 duo
1111 — moved 3 — 1 unit 1 duo
111 — transferred 4 — 0 units 0 duets 1 quartet
11 — transferred 5 — 1 units 0 duets 1 quartet
1 — transferred 6 — 0 units 1 duet 1 quartet
— transferred 7 — 1 unit 1 duet 1 quartet

Now we remember that the most significant digit comes first, we add the designation 0b to indicate it is binary notation, and we get the result: 0b111 = 7 in decimal.

If in the decimal number system we are interested in units such as tens, hundreds, thousands, and so on, in fact 10 multiplied by itself several times, then in the binary number system, we are interested in the powers of two 2, 4, 8, 16, etc.

Let's get back to Metaconformer. It displays binary numbers, where the units are located on the left side of the LEDs, and each subsequent digit is 1 position to the right. If the LED is off, it is 0, and if it is on, it is 1.

For example, what is the binary number 0101 displayed on Metaconformer equal to in decimal notation?

0 ones, 1 duet, 0 quartets, 1 octet, i.e. 2 + 8 = 10

and the number 0b0111 would be, 0 units, 1 duet, 1 quartet, 1 octet = 2 + 4 + 8 = 14.

Hopefully, now it's easier for you to navigate between binary and decimal numbers. This knowledge should be enough to successfully interact with Metaconformer.

**MIDI Protocol**

Knowing how binary numbers work, you can easily understand the structure of the MIDI protocol. Why there are 128 notes, why there are also 128 levels of velocity, 16 channels, and only 8 types of commands.

Most MIDI messages are 3 bytes long. A byte is a number with 8 bits. Here are some powers of two.

0b1 = 1
0b10 = 2
0b100 = 4
0b1000 = 8
0b10000 = 16
0b100000 = 32
0b1000000 = 64
0b10000000 = 128

The smallest number that can be written in a byte is

0b00000000 = 0

The largest number that can be written in a byte is

0b11111111 = 128+64+32+16+8+4+2+1 = 255.

256 will no longer fit in a byte, since all bits will go from 1 to 0 and it will take the 9th bit to store 1 in it.

So a MIDI message is 3 bytes. The first byte is called the status byte because it stores the type of message and the number of the channel to which the message is intended. The second and third bytes store the recipient's parameter settings.

What does "a byte stores the message type" mean? Just like with writing numbers, this is just an agreement between people on how we will interpret certain numbers.

0b10010111 — what does this byte mean? Without social agreement, it will simply denote the number 153 (make sure I'm not mistaken).

But in the MIDI protocol, people have agreed that the upper four bits of the status byte store the command number, and the lower ones store the channel number. Thus, people agreed to write in one 8-bit number, two numbers of 4 bits each.

These numbers are called nibbles. It turns out we have the ability to encode a total of 16 channels and 16 types of commands.

But is that all? There are two more data bytes. How to distinguish the status byte, which generally says WHAT we will configure, from the data byte, which says HOW we will configure.

In MIDI, the most significant bit of the byte is responsible for this. If it is equal to 1, then this is the status of the bytes. If it is 0, then it is a data byte.

This bit essentially works in general as a toggle switch that distinguishes between two types of bytes.

It turns out that we have just used up 1 bit from each byte, simply for encoding the type of the byte, but it says nothing about the data inside it. This means that only 128 values can be encoded in one byte of the message.

The general message looks like this:
0b1TTTCCCC 0b0DDDDDDD 0b0DDDDDDD, where
TTT — is 3 bits specifying the command type
CCCC is 4 bits specifying the channel number
DDDDDDD — 7 value bits, depending on message type

In the case of the status byte, this indicates that the number of channels remains 16 (after all, they sit in the low nibble), but the number of commands has decreased from 16 to 8, which also turns out to be sufficient in practice.

**MIDI message types**

**NoteOff**
0b1000_CCCC 0b0NNN_NNNN 0b0VVV_VVVVV

**NoteOn**
0b1001_CCCC 0b0NNN_NNNN 0b0VVV_VVVVV

**Polyphonic Aftertouch**
0b1010_CCCC 0b0KKK_KKKK 0b0VVV_VVVV

**Control Change**
0b1011_CCCC 0b0NNN_NNNN 0b0VVV_VVVV

**Program Change**
0b1100_CCCC 0b0NNN_NNNN

**Channel Aftertouch**
0b1101_CCCC 0b0VVV_VVVV

**Pitch Bend**
0b1110_CCCC 0b0LLL_LLLL 0b0HHH_HHHH

**Messages common to all channels**

**System Exclusive**
0b1111_0000 0b0VVV_VVVV 0b0VVV_VVVV ... 0b0VVV_VVVV 0b0VVV_VVVV 0b1111_0111

**Song Position Pointer**
0b1111_0010 0b0LLL_LLLL 0b0HHH_HHHH

**Tune Request**
0b11110110

**End Of Exclusive**
0b1111_0111

**Transport Messages**

**Timing Clock**
0b1111_1000

**Start**
0b1111_1010

**Continue**
0b1111_1011

**Stop**
0b1111_1100

**Active Sensing**
0b1111_1110

**Reset**
0b1111_1111

**Compression**

Existing MIDI controllers and sequencers can send messages in a compressed format, where fewer bytes are required to send the same information to the channel.

Let's say there is a sequence of messages with a chord whose 4 notes are first pressed and then released after a while. The channel will transmit 24 bytes:
noteOn_ch5 note50 vel100
noteOn_ch5 note51 vel100
noteOn_ch5 note52 vel100
noteOn_ch5 note53 vel100
noteOff_ch5 note50 vel50
noteOff_ch5 note51 vel50
noteOff_ch5 note52 vel50
noteOff_ch5 note53 vel50

The first type of compression compresses the data stream by discarding duplicate status bytes. The same information will contain 18 bytes (i.e. 25% less):
noteOn_ch5 note50 vel100 note51 vel100 note52 vel100 note53 vel100
noteOff_ch5 note50 vel50 note51 vel50 note52 vel50 note53 vel50

The same happens with sending other channel messages. Status bytes are not duplicated, but sent only when a new status byte needs to be sent.

The second type of compression compresses the stream even more and does not use the noteOff message. Instead, it sends a noteOn message with 0 velocity. Thus, it is necessary to transfer status bytes even less often. The same information will be transmitted through 17 bytes:
noteOn_ch5
note50 vel100 note51 vel100 note52 vel100 note53 vel100
note50 vel00 note51 vel00 note52 vel00 note53 vel00

This compression method helps to reduce the number of transmitted bytes by 33% at the peak. For example, you play on one MIDI channel, but nothing happens on others. In this case, you need to send the status bytes with the channel number once, and then only send data bytes — instead of 3 bytes per message, it remains to send only 2.

# TECHNICAL CHARACTERISTICS

Power . . . . . DC +5V via USB
Consumed current . . . . . 20 mA max
Dimensions . . . . . 148 x 50 x 20 mm
Weight . . . . . 0.15 kg

# CREDITS

Vadim Minkin — idea, schematics, code, design of construction.
Andrzej Slowik — EU production management and control.
Grigory Ryazanov — industrial construction for mass production.
Grzegorz Lacek — EU management and communications.
Vlad Kreimer — PCB design assistance.
Maxim Manakov — PCB design assistance.
Maria Atlasyuk — front panel design, animation.
Regina Volkova — management, communications, website.
Valeriy Zaveryaev — manual design and layout.
Thomas Lundberg — editor and proofreader.

www.somasynths.com

Vadim Minkin SOMA lab • 2023 год